

# SketchiXML: A Design Tool for Informal User Interface Rapid Prototyping

Adrien Coyette, Jean Vanderdonckt, and Quentin Limbourg

Belgian Lab. of Computer-Human Interaction (BCHI), Information Systems Unit (ISYS)  
Louvain School of Management (LSM), Université catholique de Louvain (UCL),  
Place des Doyens 1, B-1348 Louvain-la-Neuve (Belgium)  
{coyette,vanderdonckt,limbourg}@isys.ucl.ac.be  
<http://www.isys.ucl.ac.be/bchi>

**Abstract.** Sketching consists of a widely practiced activity during early design phases of product in general and for user interface development in particular in order to convey informal specifications of the interface before actually implementing it. It is quite interesting to observe that designers as well as end users have abilities to sketch parts or whole of the final user interface they want, while discussing the advantages and shortcomings. SketchiXML consists of a multi-platform multi-agent interactive application that enables designers, developers, or even end users to sketch user interfaces with different levels of details and support for different contexts of use. The results of the sketching are then analyzed to produce interface specifications independently of any context, including user and platform. These specifications are exploited to progressively produce one or several interfaces, for one or many users, platforms, and environments.

## 1 Introduction

Designing the right User Interface (UI) the first time is very unlikely to occur. Instead, UI design is recognized as a process that is intrinsically open (new considerations may appear at any time), iterative (several cycles are needed to reach an acceptable result), and incomplete (not all required considerations are available at design time). Consequently, means to support early UI design has been extensively researched [12] to identify appropriate techniques such as paper sketching, prototypes, mock-ups, diagrams, etc. Most designers consider hand sketches on paper as one of the most effective ways to represent the first drafts of a future UI [4,7,12, 15,16,17]. Indeed, this kind of unconstrained approach presents many advantages: sketches can be drawn during any design stage, it is fast to learn and quick to produce, it lets the sketcher focus on basic structural issues instead of unimportant details (e.g., exact alignment, typography, and colors), it is very appropriate to convey ongoing, unfinished designs, and it encourages creativity, sketches can be performed collaboratively between designers and end-users. Furthermore, the end user may herself produce some sketches to initiate the development process and when the sketch is close enough to the expected UI, an agreement can be signed between the designer and the end user, thus facilitating the contract and validation. Van Duyne et al. [20] reported

that creating a low-fidelity UI prototype (such as UI sketches) is at least 10 to 20 times easier and faster than its equivalent with a high-fidelity prototype (such as produced in UI builders). The idea of developing a computer-based tool for sketching UIs naturally emerged from these observations [12,17]. Such tools would extend the advantages provided by sketching techniques by: easily creating, deleting, updating or moving UI elements, thus encouraging typical activities in the design process [3] such as model-checking and revision. Some research was carried out in order to propose a hybrid approach, combining the best of the hand-sketching and computer assisted interface design, but this marriage highlights five shortcomings:

1. Some tools only support sketching activities, without producing any output: when the designer and the end user agreed upon a sketch, a contract can be signed between them and the development phase can start from the early design phase, but when the sketch is not transformed, the effort is lost.
2. Sketching tools that recognize the drawing do produce some output, but not in a reusable format: the design output is not necessarily in a format that is directly reusable as development input, thus preventing reusability.
3. Sketching tools are bound to a particular programming language, a particular UI type, a particular computing platform or operating system: when an output is produced, it is usually bound to one particular environment, therefore preventing developers from re-using sketches in new contexts, such as for various platforms.
4. Sketching tools do not take into account the sketcher's preferences: as they impose the same sketching scheme, the same gestures for all types of sketchers, a learning curve may prevent these users from learning the tool and efficiently using it.
5. Sketching tools do not allow a lot of flexibility in the sketch recognition: the user cannot choose when recognition will occur, degrading openness and when this occurs, it is difficult to return to a previous state.

In the remainder of this paper, Section 2 demonstrates that state-of-the-art UI sketching tools all suffer from some of the above shortcomings. Section 3 provides an overview of the Concrete User Interface (CUI) used in the sketching process, which results from widget abstraction. In Section 4, these widgets are recognized on demand. The multi-agent architecture of SketchiXML is outlined to support various scenarios in different contexts of use with examples. Section 5 concludes the paper.

## 2 Related Work

UI prototypes usually fall into three categories depending on their degree of fidelity, which is the precision to which they reproduce the reality of the desired UI.

The *high-fidelity* (Hi-Fi) prototyping tools support building a UI that looks complete, and might be usable. Moreover, this kind of software is equipped with a wide range of editing functions for all UI widgets: erase, undo, move, specify physical attributes, etc... This software lets designers build a complete GUI, from which is produced an accurate image (e.g., Adobe Photoshop, PowerPoint) or code in a determined programming language (e.g., Visual Basic, DreamWeaver). Even if the final result is not executable, it can still be considered as a high fidelity tool given that the result provided looks complete.